



CMPS 182: **Introduction to Database Management Systems**

Instructor: David Martin

TA: Avi Kaushik

Some Preliminaries

Administrative Stuff

- See the Syllabus for contact info, lecture schedule, grading, etc.
- Available on Piazza, Resources tab

Self-Introduction

AI R&D Scientist; CS Educator

- 20 years in Artificial Intelligence research
 - **SRI International**
 - **Nuance Communications (current)**
- 7 years software development in two startup companies
 - **Mark V Systems**
 - **Siri, Inc.**
- 3.5 years engineering management
 - **Siri team at Apple**
- Educational experience
 - Teaching University classes, professional tutorials
 - Developing online tutorials
 - Mentoring interns

Self-Introduction (2)

- **Artificial Intelligence Research**
 - **SRI International, Nuance Communications**
 - Fundamental and applied research in knowledge representation, intelligent assistance, software agents, natural language processing
 - Developed a broad range of prototype software systems
 - Led projects, authored proposals, commercialization activities
 - Contributed to technology and intellectual property underlying Siri
 - Authored over 60 peer-reviewed research articles, with over 7100 citations; edited two books
 - Senior Member, Association for the Advancement of Artificial Intelligence

Self-Introduction (3)

- **Engineering Manager, Siri Team, Apple**
 - Managed team of 5-8 members developing new capabilities in Siri's Java server-side environment
 - Responsible for over 15 Siri domains (e.g., question answering, Web search, calendar, email, messaging, microblogging, social networking, CarPlay)
 - Developed parts of a Java software system serving millions of users
 - Co-designed domains, features and conversational interactions for Siri
 - Managed technical relationships with Twitter, Microsoft (Bing), and Wolfram Alpha
 - Supervised an intern project on question answering
 - Recruited and trained team members

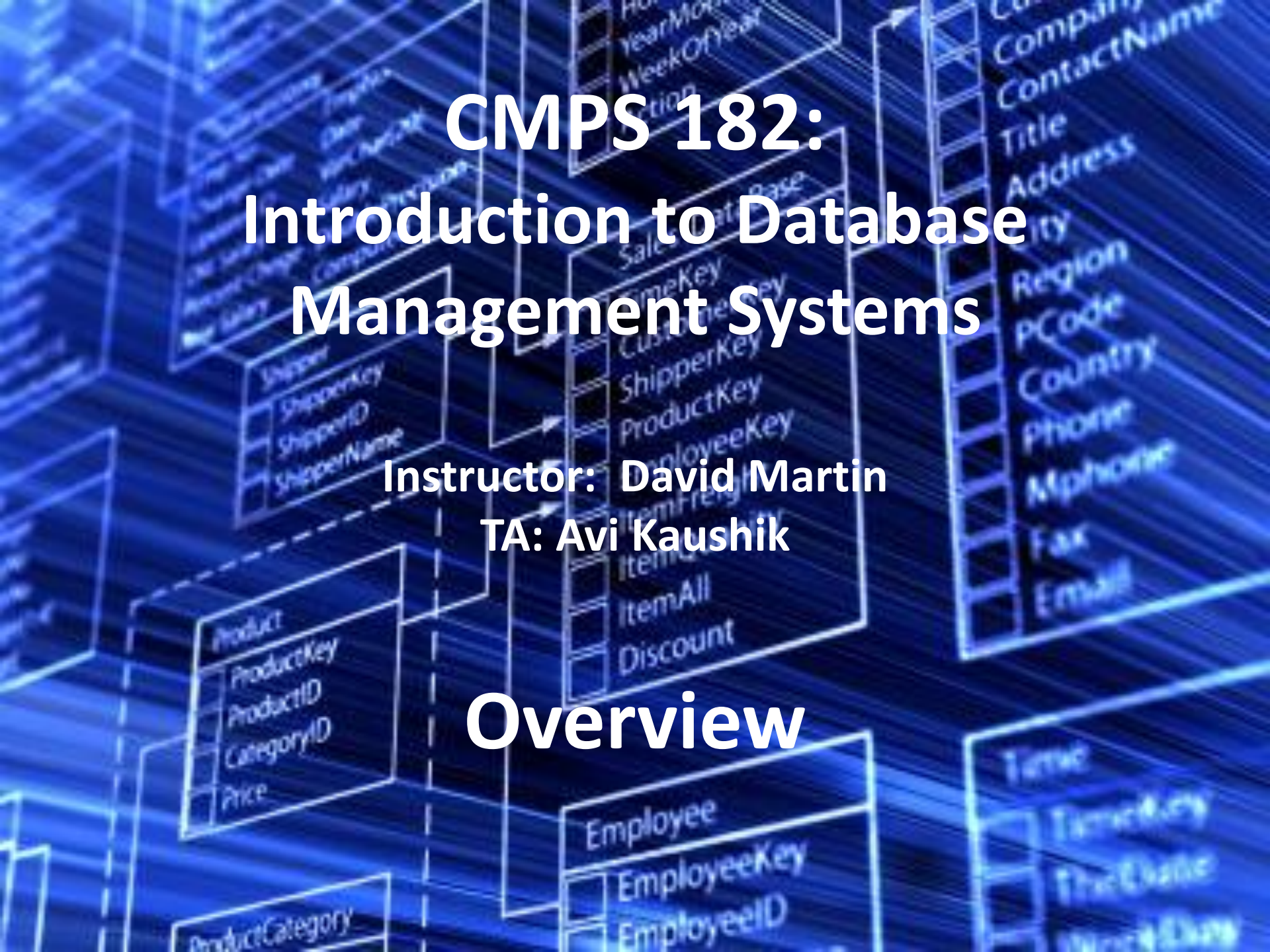
I Care About Teaching

- Some Things I'm Working On Currently
 - Better homework
 - Learning how to learn
 - Will try to have a slide or 2 each week, if possible
 - Post lecture slides before class
 - What worked; what didn't work
 - Sharing more “extra-curricular” stuff
- Please feel free to give suggestions / feedback!

Be Bold

Be Proactive

- Ask questions in class
 - Visit your professor in office hours, at least once
 - If not possible, set up a call
 - Rely on your TA; he's great!
 - Post “transparently” (non-anonymous) on Piazza as much as possible
 - Read ahead
-
- This learning opportunity is **for you**; make the most of it.
 - Making it a success is a collaborative activity **for us**.



CMPS 182: **Introduction to Database Management Systems**

Instructor: David Martin
TA: Avi Kaushik

Overview

Data...is Everywhere

MOST ENTERPRISES TODAY GENERATE
MORE DATA THAN THEY CAN PROCESS



...AND THE AMOUNT OF DATA IS GROWING AT 50% PER YEAR

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

- according to IDC

How Much Data is There?

[If There Was Already an Ocean of Data in 2007, How Much is there Now?](#), Andrew MacAfee, MIT

“There were 295 exabytes of digital data in 2007, and (by EMC estimate) [4.4 zettabytes in 2013](#), giving an annual growth rate of 57.5% over the period.”

“If we associate the volume of digital data in the world in 2007 with the volume of the Atlantic Ocean, ...

then the volume of datawater created between 2007 and 2013 would cover the Earth to a depth of 84,417 meters (276,000 feet), which is almost ten times the height of Mt. Everest.”

The Data Deluge

“... mankind created 150 exabytes (billion gigabytes) of data in 2005. “

– The Data Deluge. The Economist. Feb 2010.

“In 2011 alone, 1.8 zettabytes (or 1.8 trillion gigabytes) of data will be created, the equivalent to every U.S. citizen writing 3 tweets per minute for 26,976 years.

And over the next decade, the number of servers managing the world's data stores will grow by ten times.”

– IDC study, 2011.



From Bytes to Yottabytes

Multiples of bytes V · T · E				
SI decimal prefixes		Binary usage	IEC binary prefixes	
Name (Symbol)	Value		Name (Symbol)	Value
kilobyte (kB)	10^3	2^{10}	kibibyte (KiB)	2^{10}
megabyte (MB)	10^6	2^{20}	mebibyte (MiB)	2^{20}
gigabyte (GB)	10^9	2^{30}	gibibyte (GiB)	2^{30}
terabyte (TB)	10^{12}	2^{40}	tebibyte (TiB)	2^{40}
petabyte (PB)	10^{15}	2^{50}	pebibyte (PiB)	2^{50}
exabyte (EB)	10^{18}	2^{60}	exbibyte (EiB)	2^{60}
zettabyte (ZB)	10^{21}	2^{70}	zebibyte (ZiB)	2^{70}
yottabyte (YB)	10^{24}	2^{80}	yobibyte (YiB)	2^{80}
See also: Multiples of bits · Orders of magnitude of data				

Over the next decade, the number of **"files,"** or containers that encapsulate the information in the digital universe **will grow by**



75x

while the pool of **IT staff** available to manage them **will grow only**



1.5x

slightly.

What is a Database?

- A *database* is an organized, persistent collection of data.
 - Stored digitally.
 - Managed by a Database Management System (DBMS).
- Databases are *extremely* important.
 - Essential to every business organization.
 - Employee data, sales transactions,
 - Medical
 - Science & research
 - Web & social networking data
 - Amazon, Twitter, Facebook, IMDB, Google,
 - ...
- Databases have driven progress in Computer Science

What is a Database Management System?

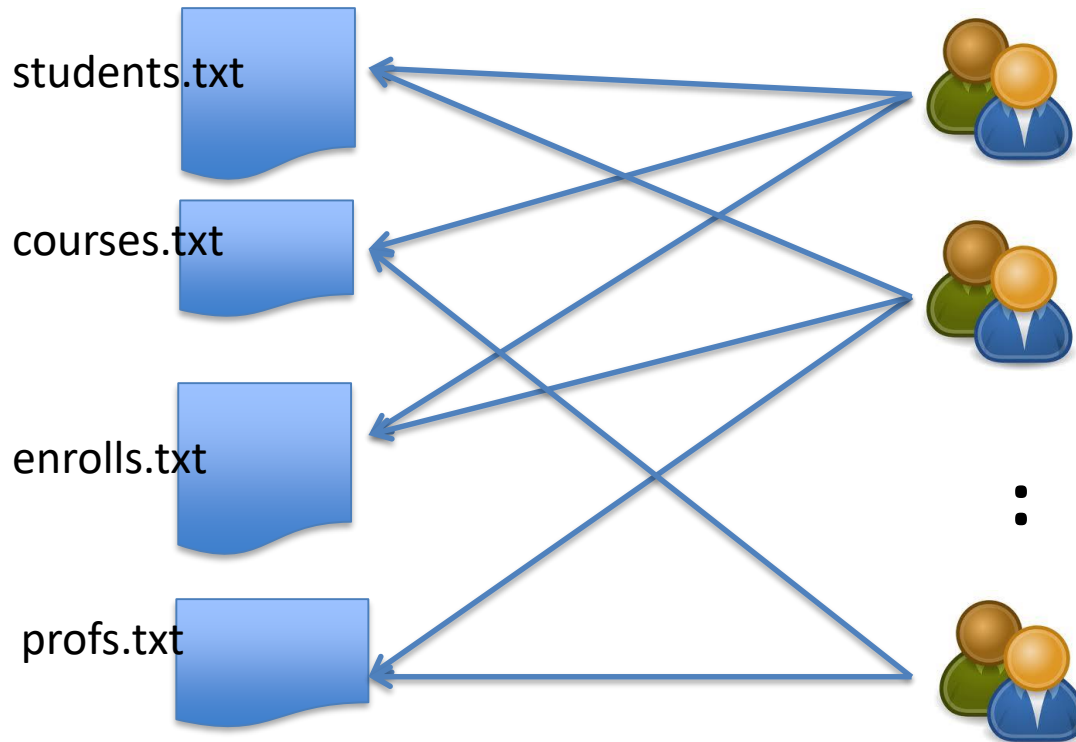
A *database management system (DBMS)* is a software system that assists in creating, maintaining, and using (typically large) datasets effectively & efficiently.

Key functionalities:

- Creation
 - Specification of *schemas* (descriptions of logical structure of data)
- Querying & modification of data
 - Asking questions to retrieve data
 - Adding, deleting & changing dataUsing a high-level language
- Storage Management
 - Long-term, scalable use of computer memory & storage resources
- Durability
- Access control & integrity

**Why can't we use a file system to
manage our data?**

Simplified version of MyUCSC Campus Portal on a file system



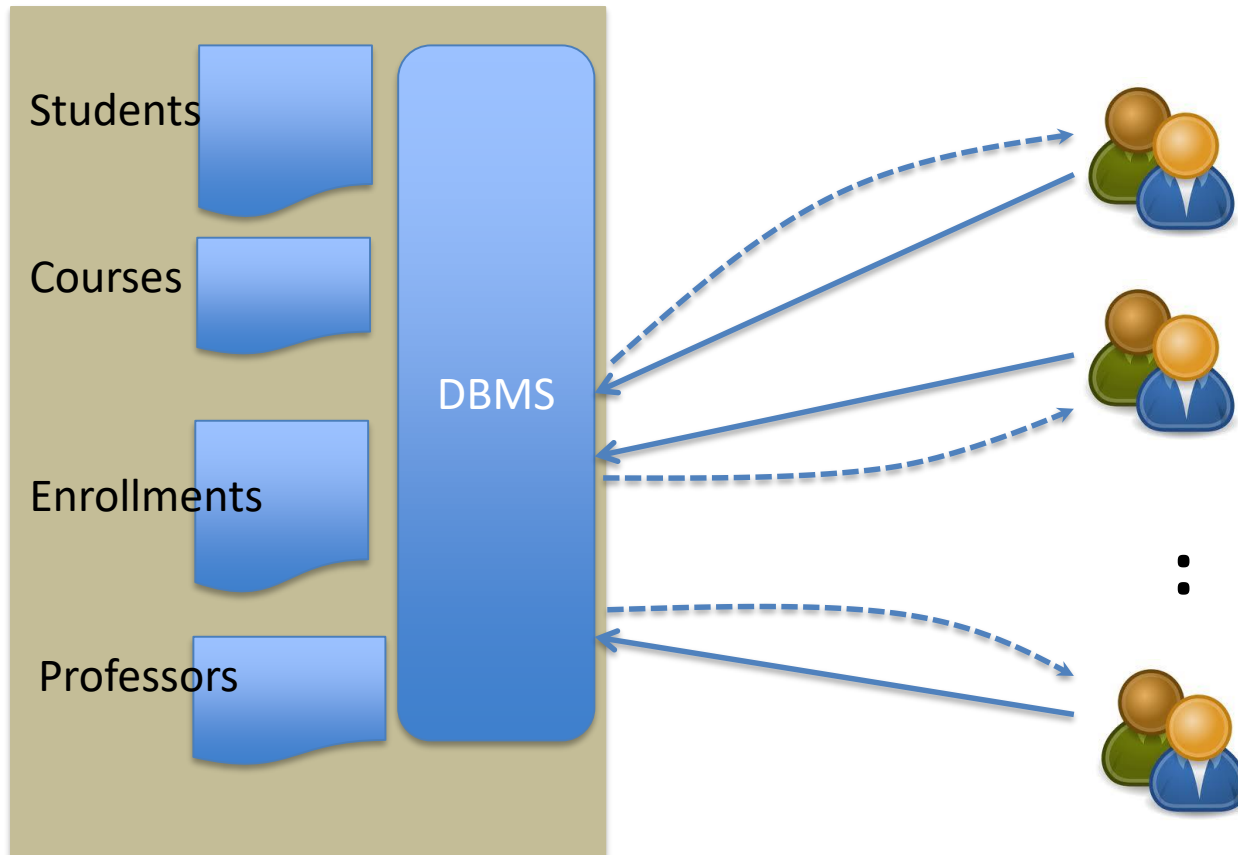
Did “Ann” get an A for
CMPS182 in Fall 2011?
Change Bob’s grade to B.

Did “Bob” enroll in
a class taught by
Prof. Kolaitis?
Change Bob’s
grade to A+.

:

Since 1960, how
many West-Coast
students got a GPA
over 3.5?

And in a Database



Did “Ann” get an A for CMPS182 in Fall 2011?
Change Bob’s grade to B.

Did “Bob” enroll in a class taught by Prof. Kolaitis?
Change Bob’s grade to A+.

Since 1960, how many West-Coast students got a GPA over 3.5?

Why can't we use a file system to manage our data?

- Visualize how an organization uses a large database.
 - Data needs to be accessed *frequently* and *concurrently*.
 - Different queries need to *posed easily* and answered *quickly*.
 - Updates to data by different users need to be managed and applied *consistently*.
 - Access to certain parts of the data by certain users need to be *restricted*.

More Key Characteristics of a DBMS

- Data Model
 - Provides an abstraction of the underlying data
 - Provides organizing principles for DBMS capabilities
 - Provides theoretical basis
- High-level language for managing data
 - For defining, updating, retrieving and processing data
- Transaction Processing
 - Concurrent access and updates, crash recovery
- Performance
 - Response time/latency
 - Throughput
 - Scalability

Transactions have the ACID properties

- A
- C
- I
- D

Transactions have the ACID properties

- A(atomicity)
- C(consistency)
- I(isolation)
- D(durability)

Advantages of a DBMS

- Users only need to understand the data model and high-level language for manipulating data.
 - Users focuses on *what* data is to be accessed and not *how* data is accessed.
 - Users are not aware of how data is actually stored or laid out on disks.
- Illusion that they are the only users of the DBMS.
- Data integrity is not compromised by system failures.
 - Deposit: $\text{Balance} = \text{balance} + 500;$
 - In parallel, a withdrawal for your monthly car payment: $\text{Balance} = \text{balance} - 300;$
 - system crashes... What is the balance?

Advantages of a DBMS (cont'd)

- Queries are automatically optimized for efficiency.
- Integrity of data is automatically enforced.
 - E.g., Employee id is unique, age < 200.
- Ease of data administration.
 - Well-developed user interfaces.
- Fast application development.
 - Available APIs and libraries.
- Data is managed centrally.
 - Costs are shared across applications.

Peak-ahead; no
need to study
at this
time

attribute names or column names

studentID	name	major	gender	avgGPA
112	Ann	Computer Science	F	3.95
327	Bob	Computer Science	M	3.90
835	Carl	Physics	M	4.00

rows or tuples

columns

3 rows, 5 columns. The relation has *arity* 5.

Peak-ahead; no
need to study
at this
time

Defining a Table

CREATE TABLE Movies (

title CHAR(100),

year INT,

length INT,

genre CHAR(10),

studioName CHAR(30),

producerC# INT

);

title	year	length	genre	studioName	producerC#
-------	------	--------	-------	------------	------------

Peak-ahead; no
need to study
at this
time

Foreign Key

Key

Sells

bar	beer	price
Above AVG	Heineken	6.00
Above AVG	Sapporo	6.00
Chugs and Slugs	Budweiser	4.00
Chugs and Slugs	Sapporo	4.00
Chugs and Slugs	Heineken	5.00
Hi Bar	Heineken	3.00
Hi Bar	Heineken	4.00
McGinty's Pub	Molson Golden	5.00
Select Clientele	Heineken	7.00

Beers

name	manf
Heineken	...
Sapporo	...
Budweiser	...
Sagres	...
Molson Golden	...

Kinds of Constraints

- **Keys/Unique** constraints
- **Foreign-key**, or referential-integrity constraints
- **Value-based** constraints
 - Constrain values of a particular attribute
- **Tuple-based** constraints
 - Relationship among components of tuple
- **Assertions**
 - Any SQL boolean expression (not implemented in most relational DBMS, not discussed much in this lecture)

Peak-ahead; no
need to study
at this
time

A Simple SQL Query

- Find all movies produced by Disney Studios in 1990.

```
SELECT *
```

```
FROM Movies
```

```
WHERE studioName = 'Disney' AND year = 1990;
```

Movies

Title	Year	Length	Genre	studioName	producerC#
Pretty Woman	1990	119	Romantic	Disney	999
Monster's Inc.	1990	121	Animation	Dreamworks	223
Jurassic Park	1998	145	NULL	Disney	675

Peak-ahead; no
need to study
at this
time

Primary Index

```
SELECT * FROM Movies
WHERE Title = Monsters, Inc.
AND Year = 1990
```

MovieIndex

Title	Year	Ptr
Alien	1979	5
Back to the Future	1985	6
Jurassic Park	1998	3
Life Is Beautiful	1997	8
Monsters, Inc.	1990	2
Pretty Woman	1990	1
Princess Mononoke	1997	7
Star Wars IV	1977	4

Primary key

Movies

Title	Year	Length	Genre	Studio	...
Princess Mononoke	1997	134	Fantasy	DENTSU	...
Monsters Inc.	1990	121	Animation	Dreamworks	...
Jurassic Park	1998	145	Adventure	Disney	...
Star Wars IV	1977	121	Sci-fi	LucasFilm	...
Alien	1979	117	Sci-fi	20 th Century Fox	
Back to the Future	1985	116	Sci-fi	Universal	
Pretty Woman	1990	119	Romantic	Disney	...
Life Is Beautiful	1997	116	Comedy	Melampo	

Binary search

Peak-ahead; no
need to study
at this
time

The Trigger

CREATE TRIGGER PriceTrig

AFTER UPDATE OF price ON Sells

The event –
only changes
to prices

REFERENCING

OLD ROW AS ooo

NEW ROW AS nnn

Updates let us
talk about old
and new tuples

We need to consider
each price change

Condition:
a raise in
price > \$1

FOR EACH ROW

WHEN(nnn.price > ooo.price + 1.00)

INSERT INTO RipoffBars
VALUES(nnn.bar);

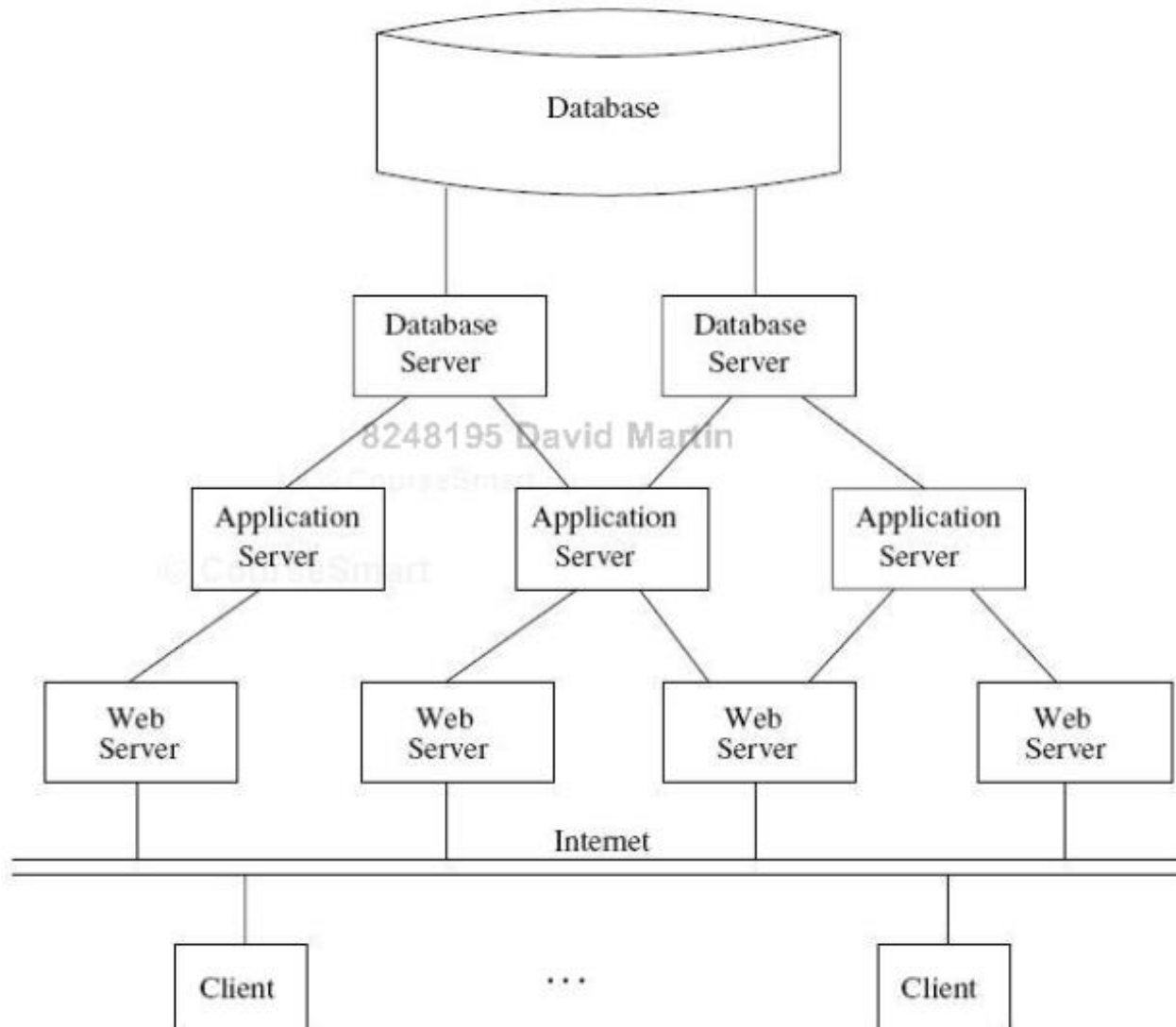
When the price change
is great enough, add
the bar to RipoffBars

Relational Algebra Operators

- Queries in relational algebra are composed using basic operations or functions on sets.
 - $\sigma_{condition}(R)$, called Selection, which is like a WHERE clause
 - $\pi_{\langle attribute\ list \rangle}(R)$, called Projection, which is like a SELECT clause
 - $R \times S$, called Product (Cartesian Product)
 - $R \cup S$, called Union (Set Union)
 - $R - S$, called Difference (Set Difference)
 - $\rho_{S(A_1, \dots, A_n)}(R)$, called Renaming

Peak-ahead; no
need to study
at this
time

3-Tier Architecture



Java: Creating Statements

- The Connection class has methods to create Statements and PreparedStatement.

```
Statement stat1 = myCon.createStatement();
```

```
PreparedStatement stat2 =
```

```
myCon.createStatement(
```

```
    "SELECT beer, price FROM Sells " +
```

```
    "WHERE bar = 'Joe' 's Bar' "
```

```
);
```

`createStatement` with no argument returns a Statement; with one argument it returns a PreparedStatement.

Peak-ahead; no
need to study
at this
time

XML and JSON, side-by-side

```
<Book>
  <Title>Parsing Techniques</Title>
  <Authors>
    <Author>Dick Grune</Author>
    <Author>Ceriél J.H. Jacobs</Author>
  </Authors>
  <Date>2007</Date>
  <Publisher>Springer</Publisher>
</Book>
```

```
{
  "Book":
  {
    "Title": "Parsing Techniques",
    "Authors": [ "Dick Grune",
                  "Ceriél J.H. Jacobs" ],
    "Date": "2007",
    "Publisher": "Springer"
  }
}
```

Peak-ahead; no
need to study
at this
time

NOSQL categories

1. Key-value

- Example: DynamoDB, Voldermort, Scalaris

2. Document-based

- Example: MongoDB, CouchDB

3. Column-based

- Example: BigTable, Cassandra, Hbased

4. Graph-based

- Example: Neo4J, InfoGrid
- “No-schema” is a common characteristic of most NOSQL storage systems
- Provide “flexible” data types

Peak-ahead; no
need to study
at this
time

Who is using them?



A Tiny Bit of History

- *Network* model
- *Hierarchical* model
- *Relational* model
- *Semistructured* model
- NoSQL models

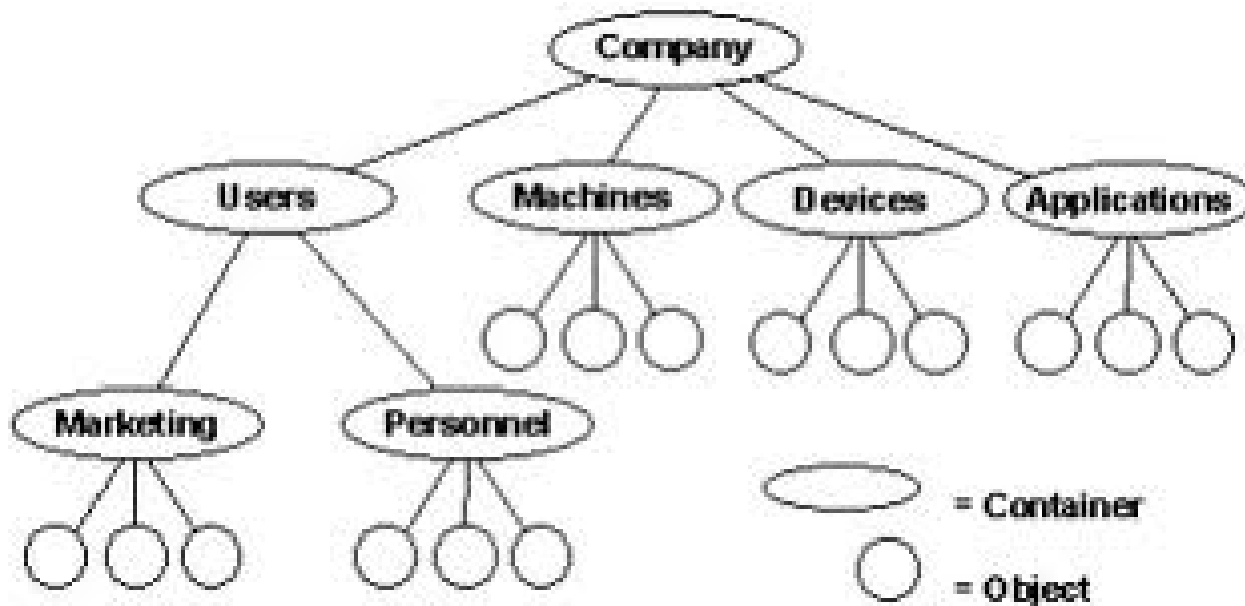
Network Data Model

- 1960s
 - First general purpose DBMS was built.
 - Integrated data store (IDS)
 - by Charles Bachman of General Electric.
 - **Network** Data Model
 - The computer navigates through a space of data records connected by pointers. A graph-based data structure.
 - A user needs to formulate the process of navigating through records and pointers to compute an answer for a query.
 - 1973 Turing award lecture.
 - “The Programmer as Navigator”



Hierarchical Data Model

- Also in the 1960s:
 - **Hierarchical** Data Model proposed at IBM (IMS product)
 - A tree-based data structure.



Relational Data Model



- 1970s
 - The beginning of *relational* database management systems.
 - Edgar (Ted) F. Codd at the IBM San Jose Research Laboratory (now IBM Almaden Research Center) published a seminal paper:
“*A relational model for data for large shared data banks*”
Communications of the ACM, 1970.

Ted Codd's Relational Model

- Advocates a radically different data model, called the *relational* data model.
 - All data must be stored in flat, table-like relations.
 - No pointers, no hierarchy !
 - Two database query languages:
 - Relational algebra and relational calculus.

Employees

EmpNo	First Name	Last Name	Dept. Num
100	Sally	Baker	10-L
101	Jack	Douglas	10-L
102	Sarah	Schultz	20-B
103	David	Drachmeier	20-B

Equipment

Serial Num	Type	User EmpNo
3009734-4	Computer	100
3-23-283742	Monitor	100
2-22-723423	Monitor	100
232342	Printer	100

Some Relational Projects and Products

- System R project started at IBM San Jose Labs in 1974.
- System R became today's DB2.
- 1981 Turing Award Lecture:
 - “Relational Database: A Practical Foundation for Productivity”.
- Michael Stonebraker and Eugene Wong at UC Berkeley started the INGRES project based on Codd's papers.
 - Relational Technology Inc. became company, INGRES
 - Later POSTGRES project led to company, Illustra Information Technologies and became open-source PostgreSQL
- Larry Ellison founded what Relational Storage Inc., which became today's Oracle Corporation. First Oracle RDBMS was released in 1979 (and was called v2)

Tidbit of the day



- Although Edgar F. Codd wrote his seminal paper at IBM San Jose Labs in 1969, and subsequently published his paper in 1970, IBM was slow to commercialize his ideas until its commercial rivals started to commercialize Codd's ideas.
 - IBM was very much into IMS/DB, an information management system based on the hierarchical data model with extensive transaction processing capabilities.
 - Used even till today by the U.S. Federal Reserve and big banks, supplemented by RDBs.
- http://en.wikipedia.org/wiki/Edgar_F._Codd

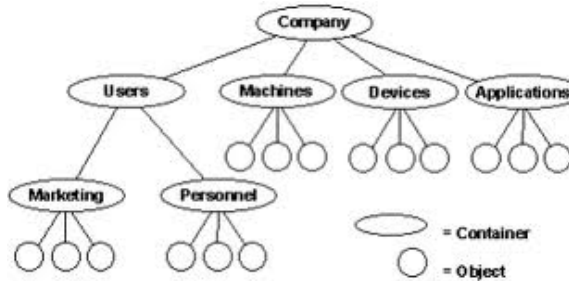
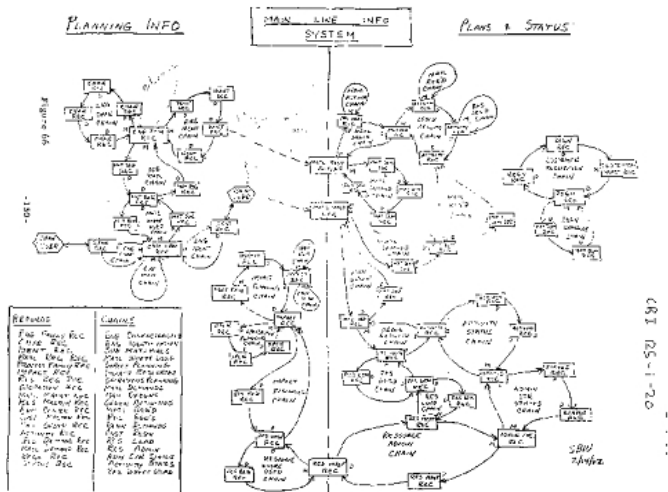
RDBMS Today

- Lots of relational database management systems
 - http://en.wikipedia.org/wiki/List_of_relational_database_management_systems
- Examples of open-source RDBMS:
 - MySQL, PostgreSQL
- Examples of proprietary RDBMS:
 - Oracle, IBM DB2, Microsoft SQL Server, SAP HANA
- Object-oriented features
- Support for semistructured sources

Major database vendors today

Relational Database Management Systems (RDBMS) Vendors					
Total Software Revenue, Worldwide, 2010-2011 (Millions of U.S. Dollars)					
Vendor	2010	2011	Share of 2010	Share of 2011	Growth 2011
Oracle	9,990.5	11,787.0	48.2%	48.8%	18.0%
IBM	4,300.4	4,870.4	20.7%	20.2%	13.3%
Microsoft	3,641.2	4,098.9	17.6%	17.0%	12.6%
SAP/Sybase	744.4	1,101.1	3.6%	4.6%	47.9%
Teradata	754.7	882.3	3.6%	3.7%	16.9%
Other Vendors	1,315.3	1,389.7	6.3%	5.8%	5.7%
Grand Total	20,746.6	24,129.5	100.0%	100.0%	16.3%
Source: Gartner (March 2012)					

1960s: Network data model and hierarchical data model



1970s: Relational data model

EmpNo	First Name	Last Name	Dept. Num	Serial Num	Type	User EmpNo
100	Sally	Baker	10-L	3009734-4	Computer	100
101	Jack	Douglas	10-L	3-23-283742	Monitor	100
102	Sarah	Schultz	20-B	2-22-723423	Monitor	100
103	David	Drachmeier	20-B	232342	Printer	100

```

{
  "photos": {
    "page": 1,
    "pages": 34276,
    "perpage": 15,
    "total": "1414129",
    "photo": [
      {
        "id": "3891667779",
        "owner": "354681331200001",
        "secret": "447960abf9",
        "server": "2451",
        "farm": 3,
        "title": "Mexican train dominoes with Brian and Michelle",
        "ispublic": 1,
        "isfriend": 0,
        "isfamily": 0
      },
      {
        "id": "3891661852",
        "owner": "106480010007",
        "secret": "79de502257",
        "server": "2590",
        "farm": 3,
        "title": "Peeches",
        "ispublic": 1,
        "isfriend": 0,
        "isfamily": 0
      }
    ]
  }
}

```

2010s: JSON

```

<Books>
  <Book ISBN="0553212419">
    <title>Sherlock Holmes: Complete Novels...
    <author>Sir Arthur Conan Doyle</author>
  </Book>
  <Book ISBN="0743273567">
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
  </Book>
  <Book ISBN="0684826976">
    <title>Undaunted Courage</title>
    <author>Stephen E. Ambrose</author>
  </Book>
  <Book ISBN="0743203178">
    <title>Nothing Like It In the World</title>
    <author>Stephen E. Ambrose</author>
  </Book>
</Books>

```

1990s: XML (Semi-structured data model)

Today, Data Also Resides Outside Enterprise Databases

- Before the Web
 - Data typically reside within enterprises, in databases
- Today
 - Data resides in enterprises and on the Web
 - Enterprise data
 - Typically sensitive information.
 - Bank accounts, employee data, sale transactions
 - Data on the Web
 - Amazon, Twitter, Facebook, IMDB, Google, ...

amazon.com



Today

In addition to RDB –

- Semistructured
 - XML, JSON
- NOSQL (“Not Only SQL”) systems
 - Map/Reduce (Hadoop)
 - Column store (HBase, Cassandra)
 - Graph databases (Neo4J, Virtuoso)
 - Semantic Web
 - Document databases (MongoDB)
- Simplicity of design, easy scale-out
- Compromise consistency in favor of availability and partition tolerance
- Lack of full ACID support, use of low-level query languages, lack of standardized interfaces (changing)

Some Supplementary Reading Material

- Database management systems
http://en.wikipedia.org/wiki/Database_management_system
- Fifty years of databases <http://wp.sigmod.org/?p=688>
- The Data Deluge. The Economist.
<http://www.economist.com/node/15579717>
- Data, data everywhere. The Economist.
<http://www.economist.com/node/15557443>

Can You Answer These Questions?

Not a homework to be submitted, but you should be able to answer these.

1. If set S is $\{1,3,5,7\}$ and set T is $\{2,3,5,7\}$, what are $S \cup T$ and $S \cap T$?
2. If set A is $\{1,2,3\}$ and set B is $\{u,v,w,x,y\}$, how many ways can you pick pairs of items, with the first from A and the second from B ? (In other words, what is the size of the Cartesian Product $A \times B$?)
3. If you have a set of employees (with names and salaries) where John makes 10K, George makes 20K, Ringo makes 30K and Paul makes 40K, give the names of the employees who make less than the average salary.
4. Write the truth-table for $p \text{ AND } q$, where p can be TRUE or FALSE and q can be TRUE or FALSE.